Predictive performance of psychological tests: is it better to use items than subscales?

Pratiwi Bunga, C.; Dusseldorp, E.; Karch, J.D.; de Rooij, M.

Computational Statistics and Data Analysis 185 (2023) 107767

Contents lists available at ScienceDirect

Computational Statistics and Data Analysis

journal homepage: www.elsevier.com/locate/csda

Predictive performance of psychological tests: Is it better to use items than subscales?

Bunga C. Pratiwi *,[1], Elise Dusseldorp, Julian D. Karch, Mark de Rooij

Leiden University, Faculty of Social Sciences, Institute of Psychology, Methodology and Statistics Unit, Wassenaarseweg 52, Leiden, 2233 AK, the Netherlands

ARTICLE INFO

Article history:
Received 25 March 2022
Received in revised form 16 April 2023
Accepted 17 April 2023
Available online 26 April 2023

Keywords:
Mean squared error
Statistical learning
Bias-variance trade-off
Cross-validation
Dimension reduction
Elastic net
Principal covariates regression

ABSTRACT

Using psychological tests to predict outcomes involves generating a prediction rule from these tests. For multidimensional tests, the standard approach to generate a prediction rule is to use the subscale scores of the test as predictor variables in a regression model to estimate an outcome value for each individual. The coefficients in this model are estimated with ordinary least squares and the predictive performance of the rule is estimated out-of-sample. Recently, studies used the separate items as predictors and estimated the regression coefficients with statistical learning methods to improve the predictive performance of these tests. However, it is unclear whether this approach is always beneficial. The aim is to identify factors that influence the decision whether to use items or subscales in a prediction rule, or letting the data decide between these two types of rules. Several statistical methods are used to derive the prediction rules: ordinary least squares, factor score regression, elastic net, supervised principal components, and principal covariates regression. Data from two empirical studies is analyzed and a simulation study is performed. Overall, results showed that, contrary to earlier findings, item rules are not always better than subscale rules. Subscale rules from elastic net often performed best.

1. Introduction

Many studies and applications use psychological tests to predict outcomes. Garnefski and Kraaij (2007) for example, created the Cognitive and Emotion regulation questionnaire (CERQ) and used it to predict anxiety and depression. Similarly, Cubiks (2018) created the Personality and Preference Inventory-Normative (PAPI-N) and used it to predict job performance. A common approach is to use the total test score as a predictor variable in a multiple regression analysis; the fitted model can be used to predict the outcome value for new individuals and is referred to as a prediction rule (Hastie et al., 2001). For a psychological test, a prediction rule consists of a weight for each item and a rule to combine the items (e.g., summation).

Psychological tests are usually designed to measure multiple constructs or dimensions. These tests, more commonly known as multidimensional tests, consist of items that can be grouped into several subscales. For example, the CERQ consists of 36 items, which are grouped into nine subscales. The standard approach to derive a prediction rule from a multidimensional test includes two steps:

* Corresponding author.
E-mail address: bunga.pratiwi@fsw.leidenuniv.nl (B.C. Pratiwi).

[1] Supplemental materials may be found on the journal page and the GitHub page of the first author, https://github.com/bcpratiwi/PredperItemsSubscales.

https://doi.org/10.1016/j.csda.2023.107767

1. Sum the items that belong to the same subscale. The item grouping is usually determined in previous research on a test's construct validity. The subscales are often based on a unit-weighted sum of a group of items.
2. Regress an outcome variable on these subscales in a multiple linear regression model using ordinary least squares (OLS).

In step one, the predefined structure of a test is used, and in step two, a prediction rule is derived from a model using the subscales as predictors. The resulting rule is referred to as a *subscale rule* in which the items within a subscale have equal weights. If step one is skipped and the individual items of the test are used as predictors, an item model is used, and a rule from this model is called an *item rule*. In this rule, the items within a subscale can have unequal weights.

Unlike a subscale rule, an item rule allows for a more proper estimation of potential unique outcome-related information of the items, which may increase the prediction strength of a test (Seeboth and Mõttus, 2018). Recent studies showed evidence supporting the use of item rules. For example, several studies showed that item rules were consistently better than subscale rules in terms of predictive performance (Seeboth and Mõttus, 2018; Putka et al., 2018). Seeboth and Mõttus (2018) found that item rules had higher predictive accuracy in predicting 39 out of the 40 outcomes they investigated. Putka et al. (2018) found that item rules performed consistently better on average. A study from Schmid and Dusseldorp (2010) showed that only one item from a 23-item subtest measuring sociolinguistic and personal background was found to be a significant predictor of verbal fluency, while the extracted subscales from the same test were not predictive.

Based on these studies, using the items directly seems to be a better approach. In spite of this, careful consideration of the statistical methods used for deriving item rules should not be overlooked. Often, OLS is used for prediction. However, if the number of items is larger than the available sample size, OLS cannot be used to derive an item rule, simply due to estimation problems. In addition, items that belong to the same subscale are likely to be highly correlated, which increases the risk of unstable estimates. Such unstable estimates can result in unstable predictions, thus leading to inaccurate predictions for new samples. The limitations of OLS can be overcome with the methods from the field of statistical learning (i.e., machine learning) such as regularized regression (Zou and Hastie, 2005; Tibshirani, 1996; Hoerl and Kennard, 1970).

Several studies have used item rules in combination with statistical learning methods. Chapman et al. (2016) found that item rules from statistical learning methods had better out-of-sample prediction accuracy than those from a generalized linear model. Seeboth and Mõttus (2018) used a regularization method to estimate item rules, knowing that OLS leads to sub-optimal coefficients for prediction due to the collinearity between the items. Putka et al. (2018) found that item rules from statistical learning methods performed better than OLS, especially in small samples.

Despite its growing usage and evidence supporting the use of item rules, it is unclear whether it is always better to ignore the standard approach and use the item rules derived from statistical learning methods. When it is unlikely that unique outcome-related information in the items is present, the payoff of using item rules instead of subscale rules may not be worth the additional effort. In addition, the grouping of the items into the subscales is originally meant to represent (an aspect of) a psychological construct, which makes the interpretation of a subscale rule easier and require less modeling effort.

The proponents of item rules would argue that they are better than subscale rules because the grouping inherent in making subscales ignores the items' unique-outcome information. Therefore, grouping items into subscales and then estimating a rule is probably not the optimal way to predict. This idea is supported by Seeboth and Mõttus (2018) who found that after gradually removing five of the most predictive items (identified by elastic net) from the corresponding subscales, the subscale rule's predictive accuracy reduced substantially. However, the reduction could also be due to the decrease in the reliability of the subscales and not due to the loss of unique item-outcome information (Seeboth and Mõttus, 2018). In sum, it is still unclear whether the standard approach is always worse than using the items.

In this study, we aim to identify the conditions in which item rules have better predictive performance than subscale rules and vice versa. Deciding beforehand whether to use items or subscales to build a rule means that we assume one approach will lead to better predictions than the other. Alternatively, we can remove this assumption by letting this choice be made in a data-driven way. Thus, we will also investigate in which situations it is better to let the data decide the type of rule. To understand why item and subscale rules work best in certain situations, we utilize the bias and variance trade-off, a guiding principle in statistical learning. The predictive performance of a rule is judged by its prediction error, which consists of bias and variance. Understanding how to trade off bias and variance of a prediction rule is an important skill to develop well-performing item and subscale rules.

Two studies have previously compared subscale and item rules but only on real data (Putka et al., 2018; Seeboth and Mõttus, 2018). Analysis on real data gives more attention to how variance plays a role in prediction error and less to how bias (i.e., how a rule is close to the true model) plays a role. In our study, we compare item and subscale rules not only on real data but also on simulated data. Studying these rules on simulated data gives more insight into the behavior of the prediction rules and more insight into how the bias and variance trade-off works as we have control over the data characteristics and the true model in the population.

As a starting point, we first discuss the bias-variance trade-off and expected prediction error. We then describe the methodology used in this study. Third, we evaluate and discuss the prediction rules on simulated data and real data. Finally, we propose practical suggestions, limitations, and future directions.

2. Bias-variance trade-off and expected prediction error

The process of deriving a prediction rule that performs well in future samples involves balancing the trade-off between bias and variance (Hastie et al., 2001). We usually estimate an outcome variable denoted as Y as a function of a random variable X with a multivariate distribution (X represents items or subscales). $f(X)$ is the true rule in the population. In general, we estimate $f(X)$ with prediction rule $\hat{f}(X)$.

How well $\hat{f}(X)$ predicts in the population is formalized by the *expected prediction error*

$$\text{EPE} = E_{X,Y}\left[\left\{Y - \hat{f}(X)\right\}^2\right], \tag{1}$$

which is the mean squared error of $\hat{f}(X)$ (Bishop, 2006).

In practice, we deploy statistical methods to estimate a rule on a training set D (a sample of observations of one data set) and return a prediction rule, typically formalized as $\hat{f}(X; D)$. Often, we are not interested in the performance of a fixed prediction rule but rather in the average performance of rules estimated for multiple data sets from the same population. Thus, we replace the squared distance of the predictions and the true values $\{Y - \hat{f}(X)\}^2$ with its average across multiple training sets $E_{\mathcal{D}}[\{Y - \hat{f}(X; \mathcal{D})\}^2]$, where the random variable $\mathcal{D}$ represents a random training set. Inserting this into Equation (1) leads to the expected predictor error for a rule estimator, which includes the type of rule and statistical method,

$$\text{EPE} = E_{X,Y}\left[E_{\mathcal{D}}\left[\{Y - \hat{f}(X; \mathcal{D})\}^2\right]\right]. \tag{2}$$

The EPE for a statistical method can be decomposed as follows (Bishop, 2006, Section 3.2)

$$EPE = (\text{Bias})^2 + \text{Variance} + \text{Irreducible error}. \tag{3}$$

$$(\text{Bias})^2 = E_X\left[\left\{E_{\mathcal{D}}.\left[\hat{f}(X; \mathcal{D})\right] - f(X)\right\}^2\right]. \tag{4}$$

$$\text{Variance} = E_X\left[E_{\mathcal{D}}\left[\left\{\hat{f}(X; \mathcal{D}) - E_{\mathcal{D}}\left[\hat{f}(X; \mathcal{D})\right]\right\}^2\right]\right]. \tag{5}$$

$$\text{Irreducible error} = E_{X,Y}\left[\{Y - f(X)\}^2\right]. \tag{6}$$

The irreducible error part of the EPE can be shown as the lower bound of the prediction error. The reducible part of the error, also known as the Mean Squared Error (MSE) of an estimator, composes of squared bias and variance of a prediction rule. The MSE of an estimator (MSE = $(\text{Bias})^2$ + Variance), is the part that we can control. Squared bias reflects how close we are to $f(X)$ when averaging the estimated rules $\hat{f}(X; D)$, variance reflects the degree to which $\hat{f}(X; D)$ varies from training sample to training sample. In general, we may see bias as mainly affected by fitting an incorrect model and variance as mainly affected by sampling variation. EPE is the sum of these two aspects plus the variance of Y in new observations. The EPE is a theoretical quantity and therefore has to be estimated.

In data analysis, we estimate EPE by evaluating the predictions from a chosen $\hat{f}(X; D)$. A chosen $\hat{f}(X; D)$ can have different compositions of bias and variance, for example, high bias and low variance or low bias and high variance. Choosing $\hat{f}(X; D)$ involves knowing how to trade off its bias and variance. For this purpose, it is helpful to formulate several expectations under which rules would perform better. These expectations depend on the true model in the population:

1. A subscale population model is defined as

$$f(X) = b_0 + \sum_{k=1}^{K} b_k \mathbf{s}_k, \tag{7}$$

where $\mathbf{s}_k$ denotes the k^{th} subscales. Subscales are defined as

$$\mathbf{s}_k = \sum_{j=1}^{J} \mathbf{x}_{jk}.$$

$\mathbf{x}_{jk}$ is the j^{th} item of subscale k and J is the total number of items per subscale. In this situation, estimated subscale rules have no bias because they follow the true model. Furthermore, a subscale rule can be seen as an item rule with a restriction that items of the same subscale have equal weights. Therefore an item rule also has no bias like the subscale rules. However, item rules might have more variance than subscale rules because of the larger number of parameters to estimate.

2. An item population model is defined as follows

$$f(X) = b_0 + \sum_{j=1}^{J}\sum_{k=1}^{K} b_{jk}\mathbf{x}_{jk}, \tag{8}$$

Note that $b_{jk} \neq b_{j'k}$ for several items belonging to the same subscale. In this case, estimated item rules have zero bias and therefore perform better than subscale rules. However, an item rule has more or equal variance than subscale rules depending on the N to P ratio. For example, for a test with 83 items grouped into 12 subscales, with a sample size of 100, an item rule could yield worse predictions than a subscale rule because the N to P ratio is larger in the item rules. However, as N increases, the difference in the ratio between the two rules eventually diminishes and item rules become better than subscale rules (see OLS rules in Putka et al., 2018).

The bias and variance of a rule are not only influenced by which predictors are in the model but also by the statistical methods used to estimate the model. OLS is known to have no bias provided the usual assumptions are met. Statistical learning methods such as elastic net (Zou and Hastie, 2005), allow control over the variance of a prediction rule. The key lies in a penalty parameter that is set by the user to control the balance of bias and variance. In general, rules from elastic net have more bias than OLS, however, this bias gets traded off with lower variance that can result in lower prediction error.

Overall, any $\hat{f}(X; D)$ will have some degree of bias and variance, which means it can take on many forms. This seems like a daunting task as there are many rules and statistical methods to choose from.

3. Methodology

3.1. Prediction rule estimation

Developing a prediction rule from a psychological test involves an interplay between the choice of using the items or subscales (or both) and statistical methods. In this study, we consider both items and subscale rules. As for the choice of methods, a natural first choice is OLS. Another method we evaluate is elastic net, which modifies the loss function of OLS. Elastic net is applied to both item and subscale rules as was previously done in Putka et al. (2018) and Seeboth and Mõttus (2018).

Apart from OLS and elastic net, we consider three approaches bearing similarities with using subscales for prediction: Factor Score Regression (FSR; Skrondal and Laake, 2001; Devlieger et al., 2016; Devlieger and Rosseel, 2017), Supervised Principal Components (SPCA Bair et al., 2006) and Principal Covariates Regression (PCovR; Vervloet et al., 2015; De Jong and Kiers, 1992). In FSR, we group the items in the same way as in the subscales, thereby theory-driven. The only difference is that the items can have unequal weights that form the factor scores, of which the weights are obtained by performing a confirmatory factor analysis (CFA) on the items. In SPCA and PCovR, the grouping of the items into subscales is data-driven by not only taking into account the covariance matrix of the items (performing a principal components analysis (PCA) or CFA on only the items) but also taking into account the items' association with an outcome variable.

SPCA and PCovR find a set of components that summarize the items and use them for prediction. These methods differ from using the subscales and factors in the following way. First, the number of components retained from the results of these methods may be different from the original number of subscales. Second, the composition of which items load highly on which components may differ from results of a PCA that was performed only on the items.

A description of the statistical methods and their implementation in R (R Core Team, 2020) is provided in the Appendix. In the next section, we formally define the prediction rules considered in this study.

3.2. Prediction rules

Let K denote the number of subscales, J denote the number of items in a subscale and Q denote the number of extracted components. In addition, let $\mathbf{y}$ be a vector of outcome scores, $\mathbf{x}$ be a vector of item scores, b denote a regression coefficient, $\mathbf{f}$ be a vector factor scores, and $\mathbf{z}$ be a vector of component scores. Component scores are defined as the weighted sum of item scores.

1. Subscale rule, where we use the subscales as the predictors in a rule

$$\hat{\mathbf{y}} = \hat{b}_0 + \sum_{k=1}^{K} \hat{b}_k \mathbf{s}_k. \tag{9}$$

2. Factor score rule, where we use factor scores as the predictors in a rule. Factors are obtained from CFA

$$\hat{\mathbf{y}} = \hat{b}_0 + \sum_{k=1}^{K} \hat{b}_k \mathbf{f}_k. \tag{10}$$

3. Item rule, where we use the items as the predictors in a rule

$$\hat{\mathbf{y}} = \hat{b}_0 + \sum_{j=1}^{J}\sum_{k=1}^{K} \hat{b}_{jk}\mathbf{x}_{jk}. \tag{11}$$

4. Component rule, where the components are the predictors in a rule. Components are derived from the items and may either be composed of items that correlate to a certain degree with the outcome (i.e., SPCA on items) or of all the items (i.e., PCovR on items)

$$\hat{\mathbf{y}} = \hat{b}_0 + \sum_{q=1}^{Q} \hat{b}_q\mathbf{z}_q, \tag{12}$$

where $\mathbf{z}_q$ is the q^{th} principal component extracted from the items, and Q is the total number of components extracted from the results of either SPCA or PCovR. Note that component rules can also be considered item rules as the items can have unequal weights.

5. Data-driven rule, estimated by a meta-method with elastic-net, which treats the input variables (i.e., items or subscales) as an additional hyperparameter (see Appendix). The algorithm will select a final rule with the lowest prediction error estimated using ten-fold cross-validation (CV). By removing the assumption of which scores are predictive, bias is reduced. However, variance increases because another parameter needs to be selected.

3.3. Predictive performance

In general, we can create eight prediction rules in this study: item rules from OLS and elastic net, factor rules from FSR, component rules from SPCA and PCovR, subscale rules from OLS and elastic net, and the data-driven rules from the meta-method. The predictive performance of OLS and FSR rules are estimated using ten-fold CV and the performance of other rules using nested ten-fold CV.

In the following we describe the difference between CV and nested CV. We begin by describing the simplest strategy to evaluate predictive performance. On a single sample, the simplest strategy is to split the data into a training and test sample. If a rule requires setting hyperparameters, we split the data into training, validation, and test samples. We fit our prediction rules in the training sample and evaluate their performance on the validation sample. One prediction rule that leads to the best performance in the validation sample is then selected. In this type of model selection, hyperparameter selection can be seen as a special case. The estimated prediction error MSE_{pr} of the selected rule in the test sample provides an unbiased estimate of EPE.

To increase the efficient use of the data, we can improve the procedure to validate and test our rules using V-fold CV, where $V = 10$ in this study. This leads to a nested ten-fold CV. In general, nested CV (with any number of folds) is used for methods with hyperparameters (Varma and Simon, 2006). Note that the choice of the number of folds V is a bias-variance trade-off problem and it is recommended to set $V = 10$ as a good compromise for this trade-off (De Rooij and Weeda, 2020). With $V < 10$, we can expect more bias but less variance, with $V > 10$, we expect less bias but more variance.

In nested ten-fold CV, we partition the data into different parts. These parts are used for model selection (validation sample) and assessment (test sample), of which these parts are usually referred to as *inner* and *outer folds*, respectively. In the inner nine folds, we perform another ten-fold CV to select a rule and train a rule on the inner nine folds and assess its performance on the left-out outer fold. The predictive performance is based on the average prediction errors of all ten outer folds. We refer to these average errors as the Mean Squared Error of Prediction (MSE_{pr}), which is an estimate of the EPE mentioned in a previous section. MSE_{pr} based on (nested) CV

$$MSE_{pr} = \frac{1}{N}\sum_{v}^{10}\sum_{i\in v}(y_{iv} - \hat{f}^{-v}(\mathbf{x}_{iv})), \tag{13}$$

where $\hat{f}^{-v}(\mathbf{X})$ denotes the predictions obtained from prediction rule estimated without the observations in fold v. The MSE_{pr} is the prediction error averaged over folds. In the empirical examples, we repeated the (nested) CV 100 times. Thus, the estimated prediction error MSE_{pr} is averaged over repetitions.

4. Monte Carlo simulation

We conducted a simulation study to obtain an overall insight into which rules performed better in various scenarios by fixing certain data characteristics and the true model in the population. The levels of the design factors were inspired by theoretical ideas on how varying data structures would affect the bias and variance trade-off and by the structure of the empirical examples, which will be described after the simulation study. The simulation study was divided into two experiments, determined by whether the true prediction model was based on component scores (experiment 1) or item scores (experiment 2).

4.1. Data generation

Let $\mathbf{x}_{jk}$ be a vector for the j^{th} item of component k. Observed item scores were generated by

$$\mathbf{x}_{jk} = v_{jk}\mathbf{z}_k + \mathbf{e}_{jk}, \tag{14}$$

where $\mathbf{z}_k$ is a vector of component scores for component k, v_{jk} is a component loading for item j for component k and $\mathbf{e}_{jk}$ is the error term for item j of component k. Component scores are collected in $\mathbf{Z}$ (matrix of N by K) and were drawn from a multivariate normal distribution $\mathcal{N}(0, \boldsymbol{\Theta})$, where $\boldsymbol{\Theta}$ denotes the covariance matrix with variances one and covariances ρ. Component loadings v_{jk} were sampled once from a uniform distribution ranging from .4 to .8 (unequal) or from .6 to .7 (more or less equal). Component loadings are collected in $\mathbf{V}$, a P by K matrix for P items and K components. We fixed the loading matrix $\mathbf{V}$ in such a way that only one component loads on a certain group of items. Error terms are collected in $\mathbf{E}$, an N by P matrix, and were generated following a multivariate normal distribution $\mathcal{N}(0, \boldsymbol{\xi})$ where $\boldsymbol{\xi}$ is a diagonal matrix with $1 - v_{jk}^2$ on the diagonal.

With the generated component matrix $\mathbf{Z}$ and observed item scores matrix $\mathbf{X}$ we simulated values for the outcome variable using

$$\mathbf{y} = \mathbf{Z}\boldsymbol{\omega} + \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon},$$

where $\boldsymbol{\omega}$ is a vector of regression coefficients of the component scores and $\boldsymbol{\beta}$ is a vector of regression coefficients of the item scores. The error term $\boldsymbol{\epsilon}$ was drawn from a normal distribution with mean zero and variance $\sigma_{\boldsymbol{\epsilon}}^2 = 1 - R^2$ and R^2 is fixed at 0.4.

We used two main variations of the above prediction model to generate the outcome variable. This variation gave us our two experiments.

- By setting $\boldsymbol{\beta} = 0$ (experiment 1), components are predictive. Note that all the items are also predictive, but their effects on Y are through $\mathbf{Z}$. We varied the percentage of signal components from 25% to 100% in increments of 25%. The percentage of signal components determined how many component scores were predictive. For example, if there were six components and 50% signal components, then the first three components were the true predictors of the outcome. The last three components had zero coefficients ($\omega_k = 0$) whereas the first three components had coefficient values determined by the following

$$\omega_k = \sqrt{\frac{R^2}{\sum_{k=1}^{K_{true}} \sigma_{Z_k}^2 + 2\sum_{k=1}^{K_{true}} \sum_{k'=1}^{K_{true}} \sigma_{Z_k Z_{k'}}}},$$

  where K_{true} is the number of components that are the true predictors of Y.
- By setting $\boldsymbol{\omega} = 0$ (experiment 2), items have direct predictive effects. The number of predictive items in the components was controlled by varying the percentage of signal items from 25 to 100% in increments of 25%. In this setting, the percentage of signal components was fixed at 50%, so half of the components had at least one predictive item. The percentage of signal items determined how many items per component had non-zero β_{jk}. For example, in the situation where there were eight components with eight items per component and 25% signal items, for each of the first four components, two of its items had equal non-zero β coefficients. These coefficients were derived using the following

$$\beta_{jk} = \sqrt{\frac{R^2}{\sum_{j=1}^{J_{true}} \sum_{k=1}^{K_{true}} \sigma_{x_{jk}}^2 + 2\sum_{j=1}^{J_{true}} \sum_{h=1}^{J_{true}} \sum_{j \neq h}^{K_{true}} \sum_{k'=1}^{K_{true}} \sigma_{x_{jk} x_{hk'}}}},$$

  where J_{true} is the number of predictive items per signal components and K_{true} is the number components that have predictive items.

In summary, for experiment 1, we had the following conditions: signal components (4), training sample size (3), number of items per subscale (2), number of components (2), correlation between components (3), and range of component loadings (2). With a fully crossed design, this gave 288 conditions. Note that the number of components manipulated in this experiment also indicates the number of subscales in a test. In this design, we were able to cover a situation where there is a small sample of 100 but a large number of subscales ($K = 20$). In addition, as we fix the component loadings, we were able to set the lower bounds of Cronbach's alpha for a single subscale to be either approximately .61 (unequal loadings) or .72 (equal loadings).

For experiment 2, we used the same 288 conditions but we varied the percentage of signal items instead of signal components. Note that the number of components also refers to the number of subscales in a test.

The simulation started with data generation for the training and test samples. For each cell of the design, we generated observed item scores and an outcome variable based on a true prediction model. We generated 100 training samples and 1 test sample of 10,000 for each cell of the design. An overview of the design factors in the simulation study is shown in Table 1.

Table 1
Design Factors for the Simulation Study.

Description	Symbol	Levels
True Prediction Models		
Experiment 1 $\boldsymbol{\beta} = 0$		
% Signal components		25%, 50%, 75%, 100%
Experiment 2 $\boldsymbol{\omega} = 0$		
% Signal items		25%, 50%, 75%, 100%
Data Characteristics		
Training Sample	n_{train}	100, 300, 500
Number of items per component	J	4, 8
Number of components	K	8, 20
Correlation between components	ρ	.3, .5, .7
Range of loadings	v_{jk}	- approximately equal (range from .6 to .7) - unequal loadings (range from .4 to .8)

4.2. Analysis

OLS item rules were excluded in this simulation study as we know that given the bias-variance trade-off, OLS item rules perform best with large training samples in which variance approaches zero. In addition, OLS item rules cannot be fitted in the condition where the number of predictors is larger than the number of observations (a condition we simulated in this study). Thus, we estimated the following prediction rules on the training samples:

1. Subscale rules from OLS
2. Factor rules from FSR
3. Item rules from elastic net
4. Subscale rules from elastic net
5. Meta-method rules
6. Component rules from SPCA
7. Component rules from PCovR

Subscales were computed as the unweighted (i.e., unit weighted) sum of the items that were generated using the same components, while factors were based on the weighted sum of the items.

4.3. Evaluation criteria

Since we generated the data in the simulation study, we used a simpler approach to evaluate the estimated prediction error MSE_{pr}. This estimate was evaluated on a large test sample of size 10,000. A uniform test sample enabled a fair comparison of the prediction rules. Note that in the training samples, ten-fold CV was performed to tune the hyperparameters.

To evaluate the performance of the rules, we first computed the percentage of conditions (i.e., 288 cells of our design) for which each rule had the lowest estimated prediction error MSE_{pr} averaged over repetitions. To identify which design factors influenced predictive performance, for each experiment, we performed two full factorial Mixed ANOVAs. The first Mixed ANOVA was performed with MSE_{pr} as the outcome variable, Rule (seven levels) as the within-subjects factor, and the other design factors that were varied in the experiments as the between-subjects factors. In the second Mixed ANOVA, we set Rule as the within-subjects factor, and OLS subscale and FSR rules were excluded.[2] We performed this Mixed ANOVA to gain more insight into the differences in performance between the statistical learning rules (i.e., elastic net, meta-method, SPCA, and PCovR).

As we were mostly interested in the difference in predictive performance between the rules, we focused on the evaluation of the within-subjects effects. These included effects of Rule and the interactions between Rule and other design factors. We only discuss highest order interaction effects with partial-eta squares $\eta_p^2 \geq .06$ indicating at least a medium effect size based on the guidelines in Cohen et al. (2013) ($\eta_p^2 = .01$ and $\eta_p^2 = .14$ are cut off values for small and large effects, respectively).

[2] FSR rules were initially included in the second Mixed ANOVA but the solution revealed no change in which of the effects were large (and the order) only that the magnitude of the effect sizes were simply lower than the effects in the first Mixed ANOVA. Therefore, we chose to show the results of the solution that also excluded FSR rules.

4.4. Some result expectations

As we know the true data generation model, we provide several expectations that are mainly based on the bias of a rule. These expectations are better understood assuming that we have a large sample because then the variance becomes small, therefore, the bias would play a more important role in a rule's predictive performance.

With regards to experiment 1 where components are predictive ($\boldsymbol{\beta} = 0$) we expect the following results.

1A Item rules from elastic net will perform similarly or worse than the other rules because elastic net item rules may shrink the coefficients of the items within a subscale (component) to zero. Thus, they have more bias.
1B Predictive performance of all rules should not depend on the percentage of signal components that are predictive in the true model because all rules would be able to capture the predictive ability of each component.
1C The difference in predictive performance between subscale rules and other rules should be influenced by the range of the component loadings. When component loadings are unequal, rules build directly from the items (i.e., item rules from elastic net, factor, and component rules) should perform better than subscale rules, simply because the subscales are based on unweighted sums (thus equal weighting) of the items. Thus, when loadings are approximately equal, subscales would perform similarly to rules from the items.
1D The performance of FSR factor, and PCovR and SPCA component rules should be comparable as they assume that the effect of items on the outcome is indirect via factors or components. In addition, component rules will have lower bias than item rules from elastic net, resulting in better predictive performance.

With regards to experiment 2 where items were directly predictive ($\boldsymbol{\omega} = 0$), we expect the following.

2A When there are 100% signal items within a component, subscale and factor rules should perform comparably with item and component rules. However, as the percentage of signal items decreases, item and component rules should perform consistently better than subscale and factor rules.
2B The difference between prediction rules should be influenced by the number of items in each component (8 or 4). When there are more items, item and component rules should perform better than subscale rules, because in subscale rules more items are being forced to have equal contributions to the prediction, which results in more bias.
2C When the component loadings are unequal and we have 100% signal items, rules from the items should perform better than subscale rules. When component loadings are approximately equal, the item and subscale models are essentially equivalent in the simulation since β_{jk}s are equal.
2D Elastic net item rules would perform consistently better than factor and component rules in cases where not all the items in a subscale are predictive. The reason is that elastic net allows for the proper estimation of the direct effects of the items on the outcome whereas factor and component rules only allow for indirect effects of the items on the outcome.

4.5. Simulation results

The results are sectioned based on the experiments. Summary tables of the results from the Mixed ANOVAs are in the Appendix in which we display the top 10 within-subjects effects (Tables A.4 to A.7). In most of the figures in this section, the scale of the y-axis was fixed (min = 0.60, max = 0.80). The minimum value of the y-axis refers to the theoretical lower bound of the prediction error in the simulated population ($var(Y) - R^2 = 1 - .4 = 0.60$). For each experiment, before we identify which design factors affect the choice between the prediction rules, we provide an overview of the percentage of conditions for which each rule had the lowest estimated prediction error MSE_{pr} averaged over repetitions.

4.5.1. Experiment 1: components are predictive

Fig. 1 visualizes an overview of the percentage of conditions for which each rule had the lowest estimated prediction error averaged over repetitions in experiment 1. As shown in this figure, out of 288 conditions, in the majority of the cases, elastic net subscale rules performed best on average. Component rules from PCovR and SPCA came in second and third. Item rules from elastic net and meta-method rules never had the lowest estimated prediction error on average in any of the cases. In a few of the conditions, FSR rules had the lowest estimated prediction error averaged over repetitions.

From the Mixed ANOVAs, we obtain insight into which situations influenced the difference in performance between the six prediction rules. In the first Mixed ANOVA, we found that (see Table A.4) the differences in average MSE_{pr} between the rules largely depended on the number of components, and training sample size, as indicated by a η_p^2 of .152 for the Rule $*$ K $*$ n_{train} three-way interaction. This interaction effect is visualized in Fig. 2. As shown in the figure, this effect was mainly driven by the difference between OLS (i.e., OLS subscale rules and FSR rules) and the other rules, especially when the number of components is 20. In this situation, OLS subscale rules and FSR rules on average performed worse than the statistical learning rules for every sample size, but the extent of this difference decreased when sample size increased, whereas when the number of components is eight, OLS subscale rules and FSR rules performed comparably with the statistical learning rules.

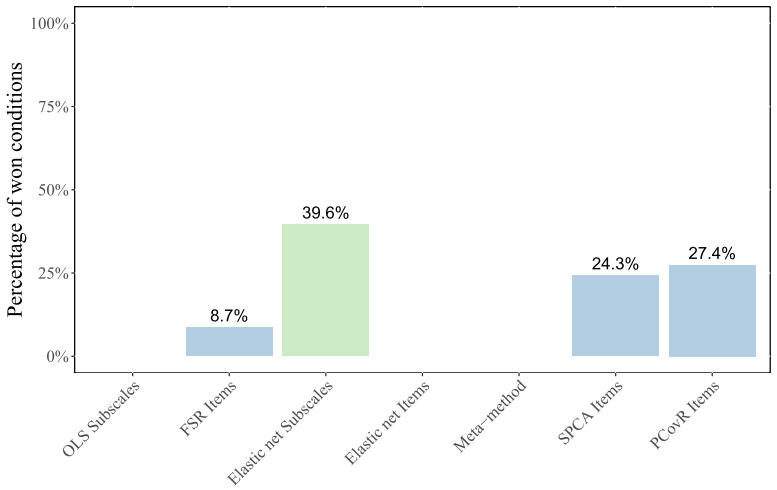

Fig. 1. Percentage of conditions (out of 288) in experiment 1 for which the rules had the lowest estimated prediction error MSE_{pr} averaged over repetitions. Colors refer to input scores: green = subscales, blue = items, and red = both (i.e., meta-method). (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

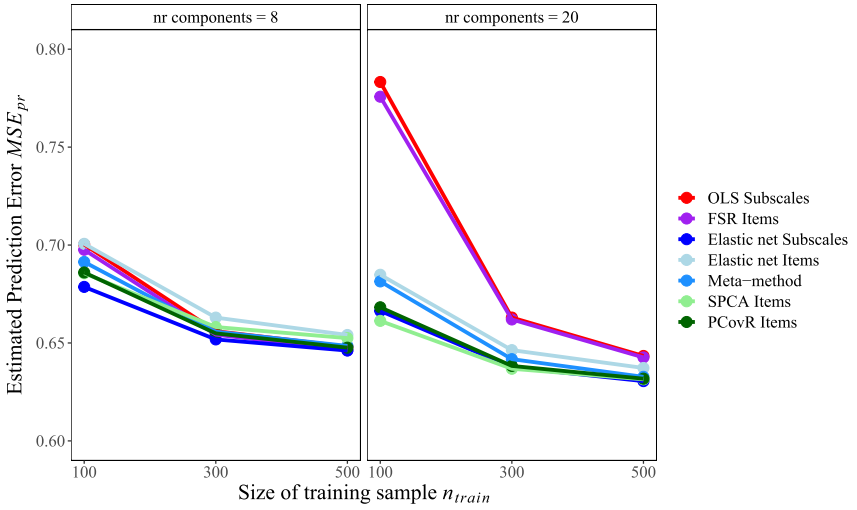

Fig. 2. Estimated prediction error MSE_{pr} in experiment 1 averaged over repetitions aggregated by training sample size and number of components K. Points refer to the mean of the estimated prediction errors. The scale of the y-axis has been fixed to compare the interaction effects on the difference in prediction errors between the six prediction rules. The minimum limit of .60 represents the theoretical lower bound of the estimated prediction error.

This three-way interaction effect partially supported our expectations. First, in regards to hypothesis 1A, when the number of components is eight, elastic net item rules on average performed as expected, that is, comparably with the other rules. However, elastic net item rules on average performed better than OLS subscale rules when the number of components was 20 while we expected elastic net item rules to perform similarly or worse.

Second, hypothesis 1B was partially confirmed as the size of the interaction effect between Rule and percentage of signal components ($\eta_p^2 = .047$) was below our threshold, but was close to it and appeared in the top five. Third, we expected that the range of the component loadings would affect the difference in predictive performance between the two types of prediction rules (hypothesis 1C). However, the interaction effect between Rule and this design factor did not make it on the top 10 within-subjects effects (Table A.4). In general, our results partially supported hypothesis 1D. SPCA and PCovR rules performed similarly but FSR rules on average performed worse than SPCA and PCovR rules. In addition, it seemed that SPCA rules on average performed slightly better than PCovR rules when there were 20 components with a small training sample of 100 (see Fig. 2).

In regards to the meta-method, the predictive performance of its rules on average were consistently in between the performance of elastic net item and elastic net subscale rules for almost all training sample sizes. For sample sizes of 500, rules from the meta-method on average performed similarly to elastic net subscale rules.

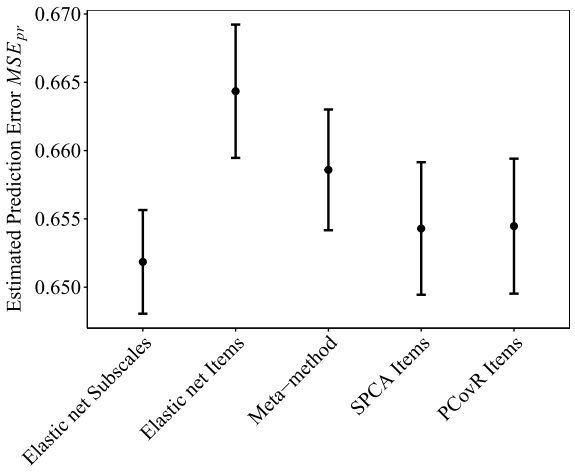

Fig. 3. Estimated prediction error MSE_{pr} of five rules in experiment 1 averaged over design factors. Points refer to the mean of the estimated prediction errors, bars refer to 95% confidence interval. The scale of the y-axis has been modified so we can compare the overall performances of the statistical learning rules.

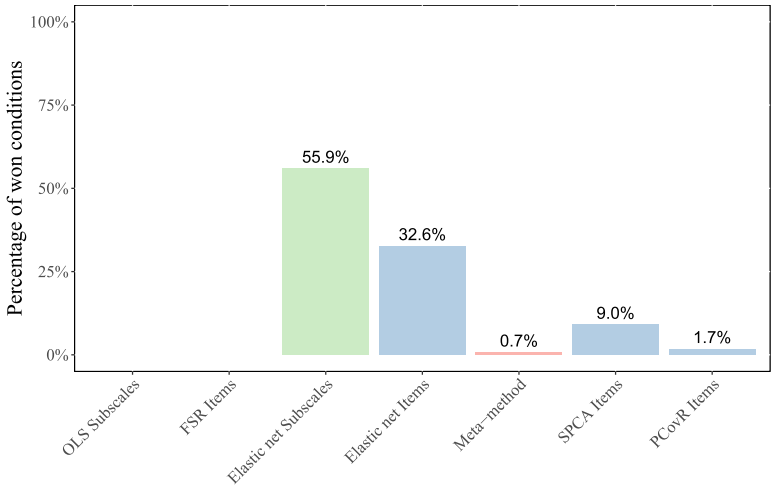

Fig. 4. Percentage of conditions (out of 288) in experiment 2 for which the rules had the lowest estimated prediction error MSE_{pr} averaged over repetitions. Colors refer to input scores: green = subscales, blue = items, and both = red (i.e., meta-method).

In the second Mixed ANOVA, after removing OLS subscale and FSR rules, the effect of Rule remained medium to large ($\eta_p^2 = .074$), while the size of all other within-subjects effects (i.e., the interactions between Rule and any of the other factors) were below our chosen threshold. The Rule $*$ K $*$ n_{train} three-way interaction did not make it onto the top 10 (see Table A.5). Fig. 3 visualizes the main effect of Rule. On average, elastic net subscale rules had the lowest estimated prediction error MSE_{pr} compared to other statistical learning rules (see Fig. 3).

4.5.2. Experiment 2: items are predictive

In this experiment, elastic net subscale rules often performed best in most of conditions averaged over repetitions (see Fig. 4), followed by elastic net item rules, which had the lowest prediction error in 32.6% of the conditions. Meta-method rules on average had the lowest prediction error in very few cases.

In the first Mixed ANOVA (Table A.6), we found that the predictive performances of the rules mainly depended on the size of the training sample n_{train}, the number of components K, and the percentage of signal items. We found that there was a large interaction effect between Rule and the percentage of signal items ($\eta_p^2 = .183$) and a large interaction effect Rule $*$ K $*$ n_{train} ($\eta_p^2 = .181$). These two interaction effects are shown in Fig. 5. In the first plot (Fig. 5a), we see that when there were 25% signal items, elastic net items on average performed best, but slightly tied with the meta-method. However, once the percentage of signal items was more than 50%, elastic net subscale rules on average performed best.

In Fig. 5b, when the number of components was 20, elastic net subscale rules on average performed best when training sample sizes were small. OLS subscale and FSR rules on average performed worst for sample sizes below 500. When the

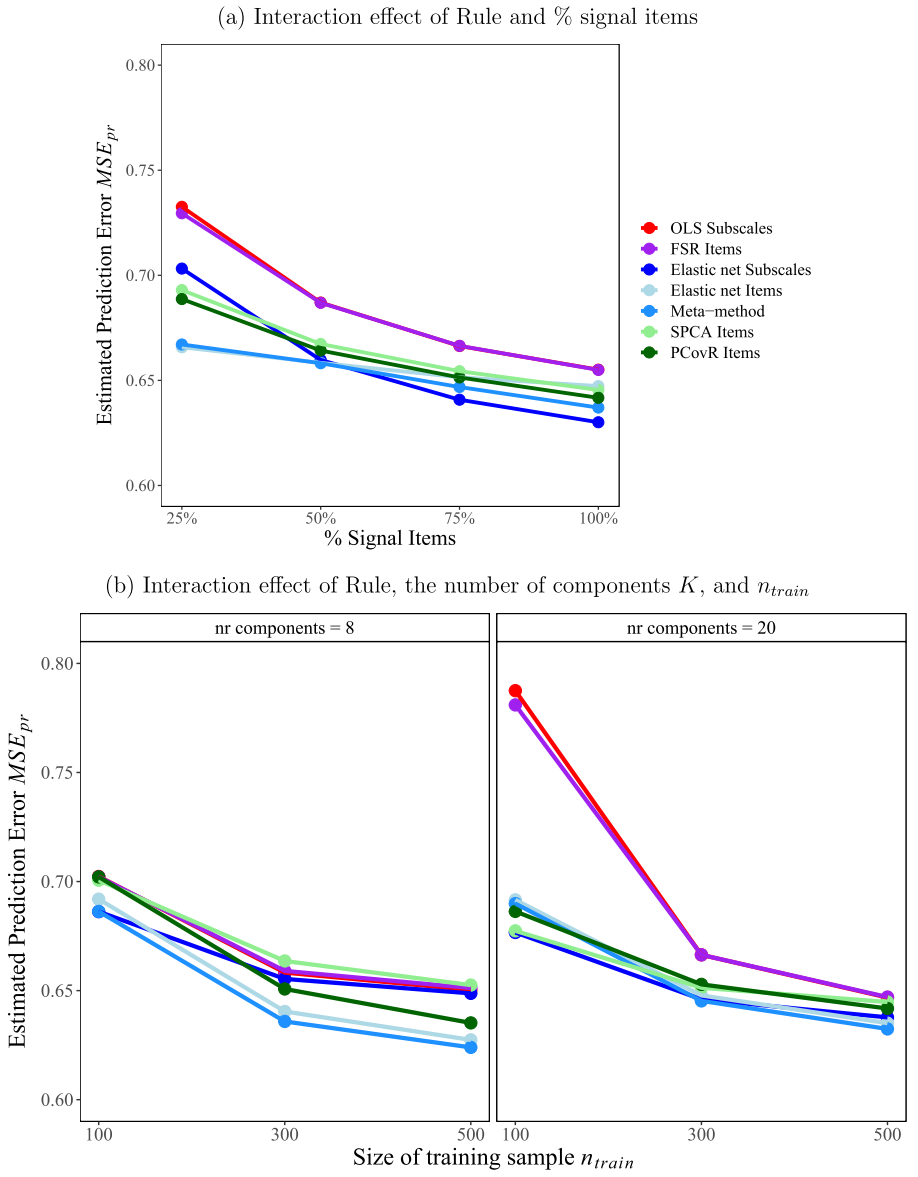

Fig. 5. Estimated prediction error MSE_{pr} in experiment 2 averaged over repetitions and other design factors that are not in the graph. Points refer to the mean of the estimated prediction errors. The colors are explained in graph a. The scale of the y-axis has been fixed to compare the interaction effects on the difference in prediction errors between the six prediction rules across figures. The minimum limit of .60 represents the theoretical lower bound of the estimated prediction error.

number of components was eight, meta-method performed best on average for sample sizes above 100. OLS subscale rules on average performed comparably with SPCA item rules and elastic net subscale rules.

These two interaction effects partially supported our expectations. In regards to hypothesis 2A, as the percentage of signal items in a subscale increased, elastic net subscale rules increasingly performed better and eventually had the best average performance when there were 100% of signal items (Fig. 5b). PCovR and SPCA component rules performed similarly across the percentages. As for hypothesis 2B, this expectation is equivalent to a three-way interaction between Rule, the number of items J, percentage of signal items, and training sample size. This effect was found to be small to medium ($\eta_p^2 = .041$). Hypothesis 2C suggests a three-way interaction effect between Rule, the range of the component loadings, and the percentage of signal items. As you can see this effect did not make it to the top 10 (Table A.6). In regards to hypothesis 2D, this expectation was met if there were less than 75% of signal items in the components (Fig. 5a). Elastic net item rules also on average performed better than factor and component rules in conditions where there were eight subscales (Fig. 5b). However, when there were 20 subscales and a small sample of 100 (Fig. 5b), PCovR and SPCA component rules on average performed better than elastic net item rules, and FSR rules performed worst on average.

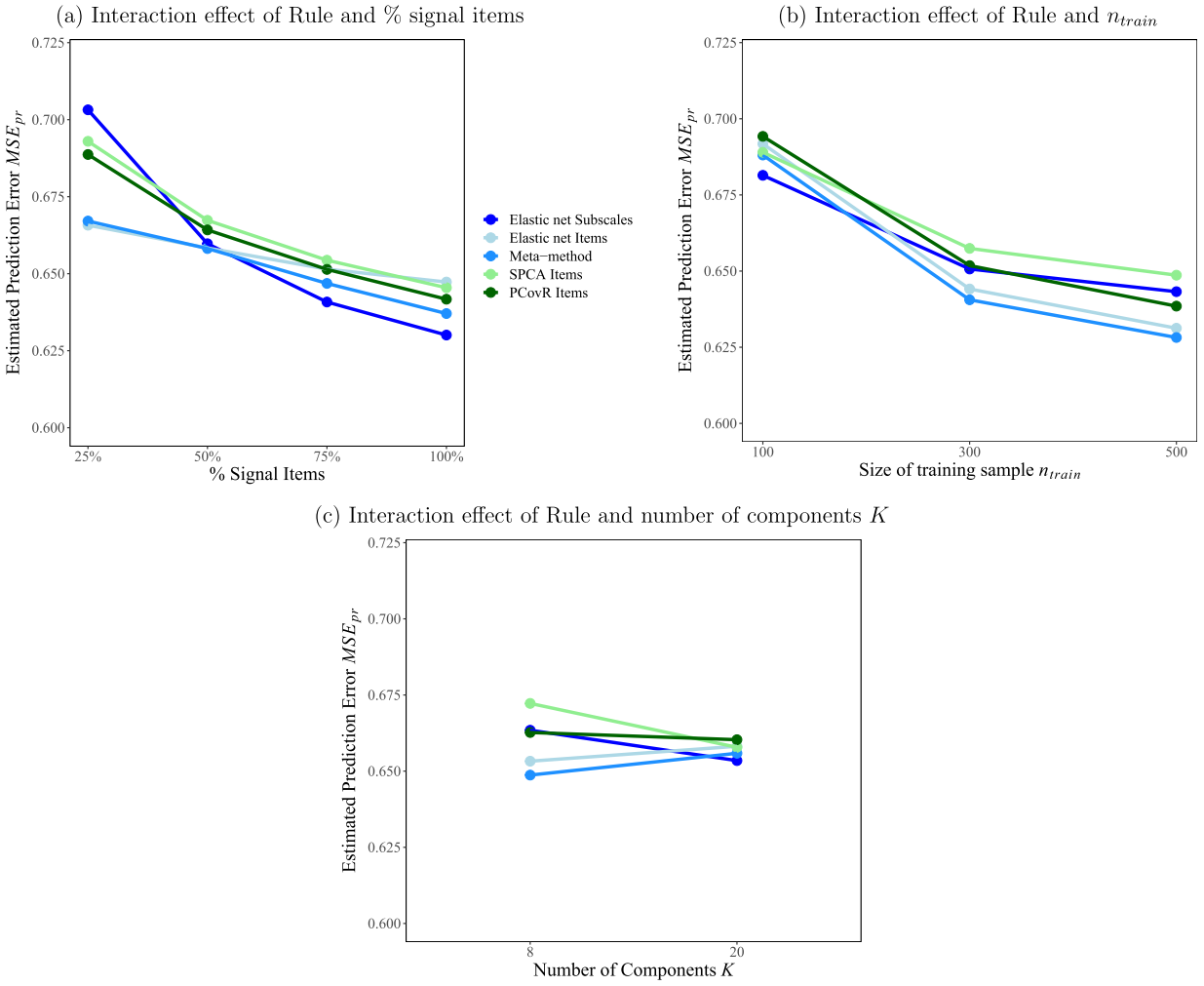

Fig. 6. Estimated prediction error MSE_{pr} of five rules in Experiment 2 averaged over repetitions and other design factors not in the graph. Points refer to the mean of the estimated prediction errors. The colors are explained in graph a. The scale of the y-axis has been modified so we can compare the predictive performances of the statistical learning rules across the figures. The minimum limit of .60 represents the theoretical lower bound of the estimated prediction error.

In the second Mixed ANOVA, we found that the interaction effect Rule $*$ K $*$ n_{train} no longer was in the top 10 within-subjects effects. However, the difference in predictive performance between the rules remained dependent either on only the size of the training samples ($\eta_p^2 = .061$) or only the number of components ($\eta_p^2 = .061$). The difference between rules from statistical learning methods remained largely depended on the percentage of signal items ($\eta_p^2 = .168$). For a summary of the results see Table A.7.

The effects from the second Mixed ANOVA (Table A.7) are visualized in Fig. 6. In the first plot of this figure, we see the same trend as in Fig. 5a where the predictive performance of elastic net subscale rules increasingly performed better on average as the percentage of signal items increased. In Fig. 5b, we see that elastic net subscale rules performed best when the training sample size is 100, while for larger sample sizes the meta-method rules performed best. In the last plot of this figure, we see that when there were eight components, the meta-method on average performed best whereas when there were 20 components, rules from elastic net, SPCA, and PCovR performed comparably.

5. Empirical examples

In this section, we present two empirical data applications in which we compare the predictive performance of several prediction rules.

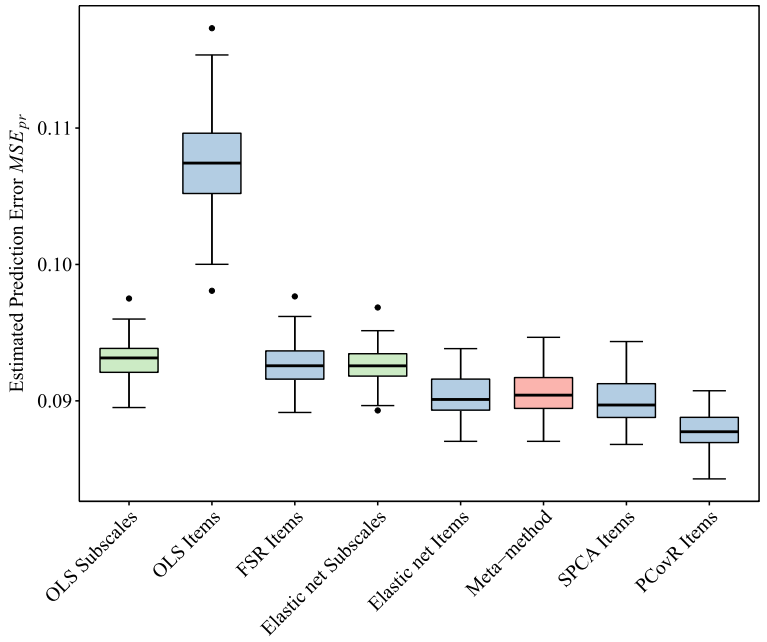

Fig. 7. Prediction Performance (MSE_{pr}) from repeated ten-fold CV for Data set 1 against prediction rules. Bold line refers to the median estimated prediction error of the 100 repetitions and the bars refer to the first and third quartiles of these repetitions. Colors refer to input scores: green = subscales, blue = items, and both = red (i.e., meta-method).

5.1. Data Set 1: description

Data set 1 came from a study on the psychometric properties of the Cognitive and Emotion Regulation Questionnaire (CERQ) in an adult general population (Garnefski and Kraaij, 2007). Items of the CERQ are aimed at measuring cognitive and emotional coping and can be used to predict depression. We used the CERQ test scores measured in the year 2000, and scores on depression measured one year later. Listwise deletion was performed to ensure that we have the same sample size to estimate item and subscale rules. As a result, the sample size was reduced from 297 to 240 in the analyses.

The CERQ consists of 36 items that form nine subscales, each containing four items. An example of an item is "I feel that I am the one to blame for it". The items are answered using a 5-point Likert scale ranging from 1 (*almost never*) to 5 (*almost always*). Scores for each subscale may run from four to 20. The subscales represent the following coping strategies: Self-blame, Acceptance, Rumination, Positive Refocusing, Refocus on planning, Positive reappraisal, Putting into perspective, Catastrophizing, and Blaming others. The Cronbach's alpha of these subscales ranged from .69 to .89 with average inter-item correlations per subscale that ranged from .36 to .67. The range of the correlation between subscales was from −.07 to .71.

The outcome variable was the sum of sixteen items measuring depression from the Dutch translation of the Symptoms Check List (Arrindell and Ettema, 1986). Items are answered on a 5-point Likert scale ranging from 1 (*not at all*) to 5 (*very much*). Original scores may run from 16 to 80. Due to a skewed distribution, the analyses were performed on a log-transformed version. These scores run from 2.77 to 4.38.

5.2. Data Set 1: analysis and results

We applied eight prediction rules for this data set: item rules from OLS and elastic net, factor rules from FSR, component rules from SPCA and PCovR, subscale rules from OLS and elastic net, and the data-driven rules from the meta-method. The predictive performance of OLS rules is estimated using ten-fold CV and the performance of other rules using nested ten-fold CV.

Fig. 7 shows the boxplots of the estimated prediction errors MSE_{pr} for the rules. As shown in the figure, using the items along with statistical learning methods (not OLS or FSR) improved the predictive performance over the standard approach of using subscales with OLS. Subscale rules from OLS and elastic net, and FSR rules tied in predictive performance. Item rules estimated from OLS yielded the highest estimated prediction error on average. PCovR component rules had the lowest prediction error on average. Prediction rules from SPCA, elastic net, and the rules from the meta-method slightly differed in their predictive performances.

In Table 2, results show that component rules from SPCA and PCovR often selected one and six components, respectively. These number of components were less than the number of subscales of the CERQ. SPCA often selected a single component solution composed of nine items. A summary of the amount of variance explained of the selected items from the SPCA

Table 2
Prediction Performance (MSE_{pr}) averaged over 100 repetitions of the prediction rules in Data set 1 sorted in ascending order with the mode of the total number of items (n_{items}) that was retained in a prediction rule and the mode of the number of predictors (P) of the best-performing rule over folds and repetitions.

Method	Input	MSE_{pr}	SD	n_{items}	P
PCovR	Items	0.088	0.001	36	6
SPCA	Items	0.090	0.002	9	1
Elastic net	Items	0.090	0.002	14	14
Meta-method	-	0.091	0.002	-	-
Elastic net	Subscales	0.093	0.001	36	9
FSR	Items	0.093	0.001	36	9
OLS	Subscales	0.093	0.001	36	9
OLS	Items	0.108	0.003	36	36

SD: computed as the standard deviation of the MSE_{pr} over 100 repetitions.
P in SPCA and PCovR are the selected number of components Q from the analysis.

Table 3
Prediction Performance (MSE_{pr}) averaged over repetitions of the prediction rules to predict overall job performance in the Data set 2 sorted in ascending order with the mode of the total number of items (n_{items}) that was retained in a prediction rule and the mode of the number of predictors (P) of the best-performing rule over folds and repetitions.

Method	Input	MSE_{pr}	SD	n_{items}	P
Elastic net	Subscales	82.837	2.199	161	27
Elastic net	Items	83.556	4.632	161	161
Meta-method	-	84.458	4.657	-	-
PCovR	Items	84.808	4.682	161	2
SPCA	Items	85.767	3.429	161	3
FSR	Items	113.970	7.808	161	27
OLS	Subscales	117.792	7.287	161	27

SD: computed as the standard deviation of the MSE_{pr} over 100 repetitions.
P in SPCA and PCovR is the selected number of components Q from the analysis.

solutions is given in Table A.8 in the appendix. The performance of the meta-method was approximately in between item and subscale rules from elastic net.

5.3. Data Set 2: description

The second data set was from a predictive validity study of the Personality and Preference Inventory-Normative Third Version (PAPI-N 3) predicting work performance measured by a multi-rater assessment (360) tool (Cubiks, 2018). Ninety-two participants completed the PAPI-N 3 in early 2014 and work performance was measured towards the end of 2015.

The PAPI-N 3 is composed of 156 items measuring personality and five items measuring social desirability. Items are answered on a 7-point Likert scale ranging from 1 (*strongly disagree*) to 7 (*strongly agree*) and can be summarized in a total of 27 subscales (26 personality subscales and one social desirability subscale). Each personality subscale (i.e., Need to influence, Need to be direct, Need to be organized, etc.) is comprised of six items. An example of an item is "I like to take an organized approach to my work.". Scores of these subscales may run from 6 to 42. Five items comprise the Social Desirability scale with scores running from 5 to 35. The average inter-item correlations per subscale ranged from .34 to .76 and Cronbach's alpha ranged from .75 to .95. In this test, the correlation between subscales ranged from $-.47$ to .66.

The outcome variable was a measure of overall job performance ratings from at least two raters measured by the 360, a Cubiks multi-rater assessment tool (Cubiks, 2018), which consisted of 50 items. Raters can answer 1 (*Substantial improvement needed*), 2 (*Slight improvement needed*), 3 (*Effective*), 4 (*Very effective*), and 5 (*Role model*). The outcome scores were based on the sum of the weighted average of three subscales in this tool. Original scores may run from 22 to 110.

5.4. Data Set 2: analysis and results

For this data set, we developed six prediction rules. Item rules from OLS were not tested since the number of predictors ($P = 161$) exceeded the sample size ($N = 92$) in this data set. Their predictive performances were evaluated in the same way as in Data set 1.

As shown in Fig. 8, prediction rules from statistical learning methods had better predictive performance than OLS subscale rules and FSR rules. Based on Table 3, elastic net subscale rules had the lowest estimated prediction error on average, but as shown in Fig. 8, their difference with the second-best rule, elastic net item rules, was small. As for the meta-method, automation did not improve predictive performance over choosing items or subscales beforehand.

As shown in Table 3, methods with variable selection such as SPCA and elastic net, often did not result in a reduction of the number of items. For SPCA, these items were often summarized into three components. In Table A.9, we provide

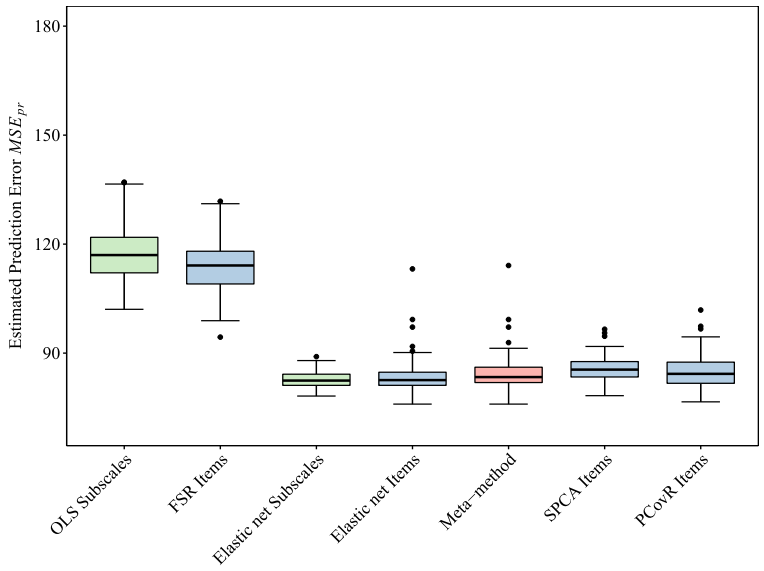

Fig. 8. Prediction Performance MSE_{pr} from ten-fold CV for PAPI-N 3 against prediction rules. Bold line refers to the median estimated prediction error of the 100 repetitions and the bars refer to the first and third quartiles of these repetitions. Colors refer to the input scores: green = subscales, blue = items, and both = red (i.e., meta-method).

a summary of the amount of variance explained by 1, 2, and 3 component solutions on the selected items from SPCA. In addition, if we compare the values of the prediction errors between elastic net subscale rules (i.e., lowest prediction error) and other rules, their predictive performances were similar as these other rules were within 1 standard deviation of the elastic net subscale rules.

Until now we evaluated the performances of several prediction rules derived from the two empirical examples. In the Appendix, we show how to obtain a final prediction and discuss the interpretation of the rules.

5.5. Conclusion empirical examples

If we assume that items contain unique outcome information in both empirical examples, we can make the following observations. First, in Data set 1, item rules from OLS suffered from overfitting, so these rules exhibited too much variance. OLS subscale rules and FSR rules performed better than OLS item rules because this variance was reduced, but they likely had more bias as they only captured the items' shared effects on the outcome. Finally, comparing OLS subscale rules and FSR rules, and prediction rules from statistical learning methods, OLS and FSR rules had more bias than the item rules from elastic net and more bias than component rules from SPCA and PCovR. Elastic net item rules and the component rules can estimate the unique item effects meaning that they had less bias and lower variance than OLS subscale rules and FSR rules.

Second, in Data set 2, OLS subscale rules and FSR rules suffered from overfitting and exhibited more bias than the statistical learning rules. In addition, these rules also most likely had more variance than the other rules. By using elastic net, the variance in the subscale rules was reduced, which lead to better predictive performance compared to OLS subscale rules and FSR rules. However, assuming that there were unique item outcome information, elastic net subscale rules probably had more bias than OLS subscale rules and other rules from statistical learning methods. PCovR and SPCA rules most likely had more variance than subscale rules as it needed to estimate more parameters (i.e., component loadings and regression coefficients of the components). However, its variance was balanced with a reduction in bias as items were allowed to have individual weights.

Furthermore, the difference in the solutions for the empirical examples can be attributed to the difference in the N to P ratio and the amount of collinearity between the items. It seemed that the improvement of statistical learning rules was less pronounced in Data set 1 than in Data set 2. We suspect that this was due to the smaller N to P ratio for subscale rules from OLS in Data set 2 (92/27) than in Data set 1 (240/9). We argue that OLS subscale rules in Data set 2 had a larger variance compared to Data set 1 due to the smaller N/P in Data 2. Therefore, the use of statistical learning methods to derive the rules was shown to substantially improve OLS-based rules (including FSR factor rules) in Data set 2 than in Data set 1.

6. Discussion

Prediction rules from multidimensional psychological tests are traditionally obtained by fitting OLS regression models on the subscales. Nowadays, researchers instead often fit a statistical learning model on the items directly, hoping that it leads

to improved predictive performance. However, it is unclear whether this is always the best approach. Thus, we compared several prediction rules that were either derived from the items, subscales, or both scores, in combination with OLS and statistical learning methods on simulated data and on two empirical examples.

We conducted a simulation study to identify the conditions in which rules derived from the items, subscales, or data-driven rules (letting the data decide between items or subscales by the meta-method) would perform best. Generally speaking, the results of the simulation suggested that the use of subscale rules from elastic net was beneficial, particularly in small samples. However, there were specific situations in which rules derived from either just the items or data-driven rules (i.e., meta-method) would be better alternatives. These situations depended on the true data-generating model, training sample size, and the number of components/subscales. In experiment 1, in which the components were predictive, SPCA component rules on average performed best when there were many components but small training samples of 100. In experiment 2, where the items have direct predictive effects and the percentage of signal items was below 75%, results indicated that elastic net item rules consistently performed better than component and subscale rules. However, the performance of elastic net item rules was often tied or was worse than the meta-method rules. When training sample sizes were large, meta-method rules often performed best on average. This suggests that when sample size is large, which means the variance becomes small, it is beneficial to minimize bias by letting the data decide which scores to use.

Contrary to expectations, in both experiments, the range of the component loadings did not influence the differences in predictive performance between subscale rules and rules build directly from the items (i.e., elastic net item rules, FSR factor rules, and component rules from SPCA and PCovR). This suggests that differentially weighting the items within a subscale was not beneficial. Thus, there is little to no gain in estimating factor scores in FSR because FSR rules had similar if not identical predictive performances with OLS subscale rules.

What is more important is the method used to weigh the subscales in a prediction rule and not how items are weighted to form the subscales. And this is mainly why elastic net subscale rules on average performed best in most conditions. There are two other plausible reasons why elastic net subscale rules outperformed other rules. First, we simulated subscales with acceptable reliabilities (i.e., Cronbach's alpha lower bounds were either .61 or.72 for unequal and equal loadings, respectively), implying that items within subscales were moderately to highly correlated. Note that these reliabilities are often encountered in practical research (e.g., CERQ questionnaire of our empirical example). By differentially weighting these items, the variance of a rule dominates the prediction error. Thus, it is beneficial to trade off this variance by increasing bias through unit weighting. For tests with weaker reliabilities (i.e., Cronbach's alpha $< .5$) or an unstable structure (i.e., different suggested subscales or grouping of the items for various samples), the advantage of unit weighting the items, which increases bias but reduces variance, would probably not compensate for the increase in bias in the subscale rules. In this situation, we expect that the bias in elastic net subscale rules would hamper predictive performance. Second, not only the number of subscales in the analysis was the same as the number of components in the population but also elastic net subscale rules always had the correct grouping. Having the same item grouping helped reduce the bias despite not using different weights to sum the items. This is unlike SPCA or PCovR which allow for different weights to sum the items but can have the wrong grouping of the items because the derived components are uncorrelated.

In both empirical examples, we found that rules from statistical learning methods outperformed OLS rules (i.e., including FSR rules). This result was consistent with the results from Putka et al. (2018) for training sample sizes below 150. In the first empirical example, item rules from statistical learning methods were better than subscale rules, a result in line with Putka et al. (2018) who found that item rules were consistently better than their subscale counterpart regardless of method, except when these rules were estimated using regression trees (i.e., CART). Seeboth and Mõttus (2018) also found that item rules were consistently better than subscale rules with both rules estimated from elastic net.

In the second empirical example, OLS subscale and FSR rules had much lower predictive performance than the other rules. This result was consistent with the finding in our simulated experiments when sample size was 100 and there were 20 components. In addition, although elastic net subscale rules yielded the lowest estimated prediction error on average, the differences in predictive performance between other statistical learning rules were small. This suggests that which statistical learning method was used did not matter. This result was similar to some of the results from Putka et al. (2018), who previously found small differences in prediction accuracy between item rules estimated using statistical learning methods except for CART.

6.1. Practical suggestions

Based on the results of this study we suggest the following. A general default is to use the item rules in combination with statistical learning methods. In many situations, these methods are able to control the variance of the item rules due to the high correlation among the items. However, in the following situations, other approaches are preferred. When sample size is small relative to the number of subscales, subscale rules should be chosen, preferably fitted with elastic net. In this setting, the variance of an item rule would probably be too high. Thus, by aggregating the items into unit weighted subscales, variance is substantially reduced from having to estimate too many parameters. With the help of elastic net, variance is reduced even more. In conditions where sample size is very large (i.e., 1000 or more), choosing between using items and subscales via the meta-method is recommended as variance will be small despite having to estimate an additional hyperparameter. Finally, in tests with a clear and stable grouping of the items (i.e., small cross-loadings and subscales with

moderate to high reliabilities) elastic net with subscales is preferred to item rules because, for the latter, the inter-item correlations will cause too much variance.

6.2. Limitations

We highlight several limitations of this study. First, we focused on linear statistical methods. Future studies should further explore nonlinear methods such as random forest and support vector machine. Note that the application of several nonlinear methods has been done on real data by Putka et al. (2018), but not for simulated data. However, it is well known that measurement error exists in psychological data and that nonlinear methods are sensitive to noise in the data, therefore these methods are more likely to overfit. Considering that we deal with moderately size data, nonlinear methods might perform worse. Second, although levels of the design factors in the simulation study were chosen to mirror real applications, they may not cover the full range of realistic data conditions. Future studies can expand them to cover more complex situations such as predicting multiple outcome variables and generating items with cross-loadings. Third, the simulations were focused on tests with a moderate to strong predefined structure. It is likely that due to this design, the unique effects of the items within subscales are less important than their shared effects for prediction. Thus, a much larger sample size is needed to properly estimate these unique effects to obtain the consistent advantage of using item rules as shown in other studies (Putka et al., 2018; Seeboth and Mõttus, 2018).

7. Conclusion

Recently, directly using the items of a psychological test to model an outcome has become a popular approach for prediction. This practice ignores prior knowledge of a group structure that exists among the items. Although this approach has been shown to provide predictive advantage (Seeboth and Mõttus, 2018), our results show the contrary. In fact, with the help of elastic net, making use of the predefined structure in a multidimensional test through subscales (unit-weighted sums of items) is recommended.

Data availability

Data set 1 and Data set 2 are subject to third-party restrictions.

Acknowledgements

We thank the following parties that have shared their data: Nadia Garnefski and Vivian Kraaij (Data set 1), and Cubiks PSI (Data set 2). We would like to thank Ethan McCormick and two anonymous reviewers for their helpful comments.

Appendix A

A.1. Statistical methods

Let P be the number of predictors K be the number of subscales, i refers to a single object or person and N denotes the sample size. In addition, let $\mathbf{y}$ be a vector of outcome scores, $\mathbf{X}$ be a matrix of predictor scores, $\mathbf{b}$ a vector of regression coefficients, $\mathbf{F}$ is a matrix of factor scores, and $\mathbf{Z}$ is a matrix of component scores.

A.1.1. Ordinary least squares

To estimate the coefficients of a prediction model, OLS uses a loss function that minimizes the squared difference between the observed values $\mathbf{y}$ and the fitted values $\hat{\mathbf{y}}$, as given below

$$L^{OLS}(\mathbf{b}) = \frac{1}{2N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 = \frac{1}{2N}\sum_{i=1}^{N}(y_i - b_0 - \sum_{p=1}^{P} x_{ip}b_p)^2. \tag{15}$$

OLS finds the regression coefficients $\mathbf{b}$ by minimizing Equation (15) on a training set. OLS produces unbiased estimates (provided the usual assumptions are met) but is known to overfit in small samples, a phenomenon caused by the fact that its estimates can vary a lot from sample to sample (McNeish, 2015). Thus, OLS has no bias but a large variance.

A.1.2. Factor score regression

Factor Score Regression (FSR) assumes that the factors are the true predictors of an outcome variable. In general, FSR contains three steps: confirmatory factor analysis (CFA),[3] factor score estimation, and prediction rule estimation using OLS.

1. Perform a CFA by fitting a factor model in the training set based on the predefined structure of the items (i.e., the grouping of the items into subscales). The fitted factor model has a simple structure, whereby each item only loads on one factor. An example is given below.

$$\mathbf{X} = \mathbf{\Lambda F} + \mathbf{e}, \tag{16}$$

where $\mathbf{F}$ is a matrix with dimension K by N of factor scores, with $K < P$ and $\mathbf{e}$ is an P by N matrix containing residual terms. $\mathbf{\Lambda}$ is a P by K matrix containing the factor loadings. Residual terms $\mathbf{e}$ are assumed to follow a multivariate normal distribution, with zero means and a diagonal covariance matrix $\mathbf{\Psi}$. Factor scores $\mathbf{F}$ are assumed to follow a multivariate normal distribution with mean $\boldsymbol{\alpha}_{(g)}$ and a covariance matrix $\mathbf{\Phi}$. Residual terms $\mathbf{e}$ and $\mathbf{F}$ are also assumed to be uncorrelated. Given the above assumptions, the model implied covariance matrix of $\mathbf{X}$ denoted by $\mathbf{\Sigma}$ is calculated as follows

$$\mathbf{\Sigma} = \mathbf{\Lambda \Phi \Lambda}^T + \mathbf{\Psi}. \tag{17}$$

The identification of the above model is either done by fixing one factor loading per latent variable to 1, or by fixing the variance of the latent variables to 1.

2. To estimate the factor scores we used the regression method from Thomson (1935). This method uses the estimated parameters $\hat{\mathbf{\Lambda}}$, $\hat{\mathbf{\Phi}}$ and the inverse of the item covariance matrix $\mathbf{\Sigma}$ as shown below

$$\hat{\mathbf{F}} = \hat{\mathbf{\Phi}} \hat{\mathbf{\Lambda}}^T \mathbf{\Sigma}^{-1} \mathbf{X}. \tag{18}$$

3. In the OLS step, we estimate a prediction rule using the factor scores in a regression model to predict an outcome variable. An example of a prediction rule is given below

$$\hat{\mathbf{y}} = \hat{\mathbf{b}} \hat{\mathbf{F}}$$

Estimation of the factor loadings does not include the outcome variable. Thus, values of the factor loadings are only affected by the covariance matrix of the items. Assuming that factors are the true predictors of an outcome, the use of factor scores -instead of subscale scores- in OLS is assumed to reduce bias because factor scores are supposedly free from measurement error. However, there can be more variance because of the factor score estimation step.

Implementation Confirmatory factor analyses and factor score estimation were performed using the developer version of `lavaan` (Rosseel, 2012). For model identification the latent variables were standardized. For model estimation, we used maximum likelihood when $N > P$ and generalized least squares with a penalized covariance matrix of the items when $N < P$ due to a non-positive definite covariance matrix. In `lavaan`, penalization of the covariance matrix was only possible for generalized least squares and the value of the penalty did not affect the size of the estimated factor loadings, therefore, we simply set the penalty to one. The regression method from Thomson (1935) (default in `lavaan`) was used to estimate the factor scores.

A.1.3. Regularization with elastic net

Regularization with elastic net prevents overfitting by adding a penalty to the OLS loss function, which involves the sum of the squared coefficients and the sum of the absolute coefficients. The loss function of elastic net is defined as

$$L^{elnet}(\mathbf{b}) = L^{OLS}(\mathbf{b}) + \lambda \left(\frac{1-\alpha}{2} \sum_{p=1}^{P} b_p^2 + \alpha \sum_{p=1}^{P} |b_p| \right). \tag{19}$$

Apart from estimating the coefficients of the predictors, the loss function has two hyperparameters: α and λ to be set. Depending on the level of α, some coefficients of the predictors will be shrunken towards zero and/or to be exactly zero. This shrinkage introduces bias but reduces variance. The α hyperparameter controls the trade-off between ridge (Hoerl and Kennard, 1970) and lasso (Tibshirani, 1996) penalty. Ridge and lasso are special cases of elastic net when $\alpha = 0$ and $\alpha = 1$, respectively. The λ hyperparameter determines how the amount of bias is introduced by the penalty. For larger λ values, we impose more bias and in turn smaller variance. When $\lambda = 0$ we end up with OLS estimates. To find the hyperparameter values that result in the best balance of variance and bias, different combinations of λ and α are used and the optimal combination is determined by cross-validation (CV).

[3] The method can also work with exploratory factor analysis.

Implementation We used R packages `glmnetUtils` (Ooi, 2021) and `glmnet` (Friedman et al., 2010) to implement elastic net. We evaluated α values from 0 to 1 in increments of .10 and a set of λs determined by default by the function. The final rule is based on a combination of α and penalty λ with the lowest MSE estimated using ten-fold CV.

A.1.4. Supervised principal components

Supervised Principal Components (SPCA) assumes that certain predictors can be summarized into a set of components and these components are the true predictors of an outcome variable. Governed by this assumption, the method requires several steps (more details in Bair et al., 2006).

1. Perform variable selection on $\mathbf{X}$ based on the bivariate relationships between predictors and outcome. This is done by fitting a univariate regression model on every predictor and outcome Y and only retaining those predictors with coefficients (absolute value) larger than threshold θ. As a result, we end up with a reduced predictor matrix $\mathbf{X}_\theta$. Note that this is akin to selecting those predictors that are correlated higher than a certain threshold.
2. Perform a PCA and select a number of components. In this study, the predictors are standardized. Thus, the PCA finds the matrix of scores $\hat{\mathbf{Z}}_\theta$ and component loadings matrix $\hat{\mathbf{A}}_{\mathbf{X}_\theta}$ that minimize

$$L(\mathbf{Z}_\theta, \mathbf{A}_{\mathbf{X}_\theta}) = \|\mathbf{X}_\theta - \mathbf{Z}_\theta \mathbf{A}_{\mathbf{X}_\theta}\|^2. \tag{20}$$

3. Estimate the prediction rule by regressing the outcome on the component score matrix $\hat{\mathbf{Z}}_\theta$ using OLS. Below is an example of a prediction rule we would obtain based on a one-dimensional solution

$$\hat{\mathbf{y}} = \hat{b}_0 + \hat{b}_1 \hat{\mathbf{z}}_{\theta,1},$$

where $\hat{\mathbf{z}}_{\theta,1}$ is the first principal component, and $\hat{b}_1$ denotes the regression weight.

Note that this method does not have a single loss function to obtain the component scores and regression coefficients.

SPCA requires the user to set two hyperparameters, the threshold θ and the number of components Q. Considering the bias and variance trade-off, if we choose a small θ, all variables may be included to form the components, which means low bias but the variance of the rule may increase. In terms of the number of components, as the number of components increases, bias decreases. However, the variance increases because not only do we allow for more flexibility in summarizing the predictor scores but also we increase the number of coefficients to estimate.

Implementation We used the R package `superpc` (Bair and Tibshirani, 2012) to implement SPCA. The hyperparameters θ and the number of components Q (typically 1, 2, and 3; Bair et al., 2006) to extract from the reduced predictor matrix are chosen based on the combination that yields the lowest prediction error estimated using ten-fold CV.

A.1.5. Principal covariates regression

Principal Covariates Regression (PCovR) is similar to SPCA but unlike SPCA, PCovR has a single loss function that simultaneously minimizes the weighted sum of the dimension reduction error and the prediction error:

$$L(\mathbf{Z}, \mathbf{A}_\mathbf{X}, \mathbf{a}_\mathbf{y}) = \gamma \frac{\|\mathbf{X} - \mathbf{Z}\mathbf{A}_\mathbf{X}\|^2}{\|\mathbf{X}\|^2} + (1-\gamma)\frac{\|\mathbf{y} - \mathbf{Z}\mathbf{A}_\mathbf{X}\|^2}{\|\mathbf{y}\|^2}. \tag{21}$$

$\|\mathbf{W}\| = \sum_{i,p} w_{ip}^2$ is the Frobenius norm of matrix $\mathbf{W}$. $\mathbf{Z}$ are the component scores, $\mathbf{A}_\mathbf{X}$ are the component loadings, and $\mathbf{a}_\mathbf{y}$ are the regression weights. The first term in this equation is responsible for the dimension reduction of $\mathbf{X}$, whereas the second term focuses on the prediction of the outcome variable Y given the components. To identify the parameters, the method sets $\mathbf{Z}'\mathbf{Z} = \mathbf{I}$, where $\mathbf{I}$ is an identity matrix of an appropriate size. It thus assumes that the components are orthogonal and that all components have variance one. For a given set of estimated parameters $\hat{\mathbf{Z}}$, $\hat{\mathbf{A}_\mathbf{X}}$, $\hat{\mathbf{a}_\mathbf{y}}$, the prediction rule is

$$\hat{\mathbf{y}} = \hat{\mathbf{Z}}\hat{\mathbf{a}}_\mathbf{y}.$$

Two hyperparameters need to be set for this method, weighting parameter $\gamma \in [0, 1]$ and the number of components Q. The hyperparameter γ controls which focus has more weight; when $\gamma = 0$, the solution is similar to an OLS regression when $\gamma = .5$, PCovR solution bears similarities with partial least squares (see De Jong and Kiers, 1992), and when $\gamma = 1$, the solution is equivalent to principal components regression. For interested readers, see Heij et al. (2007) for a comparison between PCovR and principal components regression. Any value of γ between 0 and 1 gives a compromise between PCA and regression. Note that the method does not result in variable selection.

Given a fixed number of components, as $\gamma \to 1$, we expect less bias as we impose fewer assumptions in the structure of the predictors. In terms of the number of components Q, the more components we retain, the less bias we have, but we expect more variance as we need more parameters to estimate in $\mathbf{a}_\mathbf{y}$.

Implementation We used the `PCovR` package (Vervloet et al., 2015) to implement this method. Item scores and outcome values were standardized before the analysis. To compute the estimated prediction error MSE_{pr}, the outcome variable is rescaled back to the original. We evaluated several number of components (default option is from 1 to $P/3$; Vervloet et al., 2015). In the simulation study, for efficiency, we explored 20 possible values in $.05 \leq \gamma \leq .99$. These values were used as well in Vervloet et al. (2016). In the empirical data examples, we evaluated γ values that ranged from .01 to .99 in increments of .01. The chosen rule is based on the combination of γ and the number of components that yield the smallest prediction error estimated using a ten-fold CV.

A.2. Mixed-ANOVA tables

Table A.4
Top 10 within-subjects effects sorted in descending order from a Mixed ANOVA on all seven prediction rules in experiment 1.

	SS	df	F	η_p^2
Rule	27.239	3.424	12856.383	0.311
Rule x n_{train}	25.780	6.848	6083.894	0.299
Rule x K	20.660	3.424	9751.339	0.255
Rule x n_{train} x K	14.308	6.848	3376.680	0.192
Rule x signal comps	3.048	10.272	479.501	0.048
Rule x ρ	1.062	6.848	250.626	0.017
Rule x ρ x signal comps	0.400	20.545	31.439	0.007
Rule x n_{train} x signal comps	0.292	20.545	23.005	0.005
Rule x J	0.255	3.424	120.325	0.004
Rule x n_{train} x ρ	0.251	13.696	29.648	0.004
Error (Rule)	60.409	97628.159		

df: Greenhouse-Geisser correction.
All displayed effects are significant at $p < .001$.

Table A.5
Top 10 within-subjects effects sorted in descending order from a Mixed ANOVA on five prediction rules (excluding OLS subscale rules and FSR factor rules) in experiment 1.

	SS	df	F	η_p^2
Rule	2.773	2.886	2290.983	0.074
Rule x signal comps	1.707	8.659	470.178	0.047
Rule x n_{train}	1.045	5.773	431.570	0.029
Rule x K	0.356	2.886	294.318	0.010
Rule x ρ x signal comps	0.286	17.318	39.448	0.008
Rule x n_{train} x signal comps	0.173	17.318	23.885	0.005
Rule x J	0.153	2.886	126.255	0.004
Rule x J x n_{train}	0.125	5.773	51.595	0.004
Rule x ρ	0.109	5.773	45.122	0.003
Rule x ρ x signal comps x K	0.104	17.318	14.269	0.003
Error (Rule)	34.506	82293.571		

df: Greenhouse-Geisser correction.
All displayed effects are significant at $p < .001$.

Table A.6
Top 10 within-subjects effects sorted in a descending order from a Mixed ANOVA on all seven prediction rules in experiment 2.

	SS	df	F	η_p^2
Rule	31.226	3.500	13832.976	0.327
Rule x n_{train} x K	18.949	7.001	4197.158	0.227
Rule x n_{train}	18.784	7.001	4160.639	0.226
Rule x signal items	14.539	10.501	2146.978	0.184
Rule x K	12.762	3.500	5653.758	0.165
Rule x signal items x K	3.340	10.501	493.275	0.049
Rule x J	3.015	3.500	1335.679	0.045
Rule x J x signal items	2.837	10.501	418.944	0.042
Rule x ρ	0.860	7.001	190.404	0.013
Rule x ρ x signal items	0.799	21.003	58.979	0.012
Error (Rule)	64.361	99804.326		

df: Greenhouse-Geisser correction.
All displayed effects are significant at $p < .001$.

Table A.7
Top 10 within-subjects effects sorted in a descending order from a Mixed ANOVA on five prediction rules (excluding OLS subscale rules and FSR factor rules) in experiment 2.

	SS	df	F	η_p^2
Rule x signal items	7.689	8.582	1912.883	0.168
Rule	2.823	2.861	2107.326	0.069
Rule x n_{train}	2.483	5.722	926.451	0.061
Rule x K	2.482	2.861	1852.820	0.061
Rule x signal items x K	2.022	8.582	502.939	0.050
Rule x J	1.881	2.861	1404.232	0.047
Rule x J x signal items	1.427	8.582	354.942	0.036
Rule x ρ x signal items	0.580	17.165	72.167	0.015
Rule x n_{train} x signal items	0.553	17.165	68.754	0.014
Rule x ρ	0.472	5.722	176.291	0.012
Error (Rule)	38.201	81567.508		

df: Greenhouse-Geisser correction.
All displayed effects are significant at $p < .001$.

A.3. SPCA: variance explained tables

Table A.8
Percentage of variance explained on selected items for 1, 2, and 3 component solutions from SPCA for Data set 1 over folds and repetitions.

ncomps	%selected[b]	%variance explained[a]		
		min	mean	max
1	88.2%	27.35%	42.77%	57.02%
2	8.7%	38.50%	57.17%	70.69%
3	3.1%	49.01%	65.03%	82.62%

[a] The percentage of variance explained can be based on different numbers of items.
[b] Percentage of times this solution is selected over folds and repetitions (total is 1000 solutions).

Table A.9
Percentage of variance explained on selected items for 1, 2, and 3 component solutions from SPCA for Data set 2 over folds and repetitions.

ncomps	%selected[b]	%variance explained[a]		
		min	mean	max
1	5.8%	14.01%	25.53%	51.65%
2	28.1%	22.21%	34.28%	76.44%
3	66.1%	28.06%	44.04%	92.72%

[a] The percentage of variance explained can be based on different numbers of items.
[b] Percentage of times this solution is selected over folds and repetitions (total is 1000 solutions)

A.4. Final prediction rules

For each empirical data example, we derived a final prediction rule based on the rule that yielded the lowest estimated prediction error MSE_{pr}. For further details on how these rules were derived see Supplemental Materials. For Data set 1, we derived a final rule using PCovR on the items with a varimax rotation (Kaiser, 1958). The rotation was performed to enhance interpretation.

We found that items of the CERQ can be summarized into six components, which is less than the number of the original nine subscales in the CERQ. These groupings are based on loadings that are higher than 0.4 in absolute values. The final rule for this test to predict standardized log depression scores is shown below

$$depr = 0.14Z_1 + 0.31Z_2 + (-0.04)Z_3 + 0.29Z_4 + (-0.34)Z_5 + (-0.08)Z_6.$$

We found that four of the original subscales of the CERQ were grouped into two components, items from positive coping strategies such as positive reappraisal and refocus on planning load highly on Z_1, whereas more negative coping strategies such as catastrophizing and blaming others had high loadings on Z_2. For the other components, we found that the following items had high loadings on the following components, positive refocusing on Z_3, self-blame items on Z_4, rumination items on Z_5, and acceptance on Z_6. We also found that based on the absolute value of the coefficients, Z_2 and Z_5 were the two most important components to predict depression. Component 2 had a positive weight, which means, that if a person scored high on items of negative coping strategies, they will score high on depression. Component 5 had a negative weight. Since items on rumination had high but negative loadings, this means that if a person scored high on rumination, they also will score high on depression.

For Data 2, we derived a subscale rule using elastic net. We found that all the original subscales (i.e., ridge penalty was applied in elastic net) were retained for predicting overall job performance. These effects are not interpreted further there 27 subscales in the PAPI, therefore would be verbose. However, an important takeaway from this example is that researchers can enhance the predictive performance of a subscale rule by regularizing its coefficients.

Appendix B. Supplementary material

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.csda.2023.107767.

References

Arrindell, W.A., Ettema, J., 1986. SCL-90: Handleiding Bij Een Multidimensionele Psychopathologie-Indicator. Swets Test Publishers, Lisse, the Netherlands.

Bair, E., Hastie, T., Paul, D., Tibshirani, R., 2006. Prediction by supervised principal components. J. Am. Stat. Assoc. 101, 119–137.

Bair, E., Tibshirani, R., 2012. superpc: supervised principal components. R package version 1.09 URL. https://CRAN.R-project.org/package=superpc.

Bishop, C.M., 2006. Pattern Recognition and Machine Learning. Springer, New York.

Chapman, B.P., Weiss, A., Duberstein, P.R., 2016. Statistical learning theory for high dimensional prediction: application to criterion-keyed scale development. Psychol. Methods 21, 603.

Cohen, J., Cohen, P., West, S.G., Aiken, L.S., 2013. Applied Multiple Regression/Correlation Analysis for the Behavioral Sciences. Routledge.

Cubiks, 2018. PAPI 3 SL English Language Version Manual Supplement. Guildford.

De Jong, S., Kiers, H.A., 1992. Principal covariates regression: Part I. Theory. Chemom. Intell. Lab. Syst. 14, 155–164.

De Rooij, M., Weeda, W., 2020. Cross-validation: a method every psychologist should know. Adv. Methods Pract. Psychol. Sci. 3, 248–263.

Devlieger, I., Mayer, A., Rosseel, Y., 2016. Hypothesis testing using factor score regression: a comparison of four methods. Educ. Psychol. Meas. 76, 741–770.

Devlieger, I., Rosseel, Y., 2017. Factor score path analysis: an alternative for SEM? Methodology: Eur. J. Res. Methods Behav. Soc. Sci. 13, 31.

Friedman, J., Hastie, T., Tibshirani, R., 2010. Glmnet: regularization paths for generalized linear models via coordinate descent. J. Stat. Softw. 33, 1–22. http://www.jstatsoft.org/v33/i01/.

Garnefski, N., Kraaij, V., 2007. Psychometric features and prospective relationships with depression and anxiety in adults. Eur. J. Psychol. Assess. 23, 141–149.

Hastie, T., Tibshirani, R., Friedman, J., 2001. The Elements of Statistical Learning, 1 ed. Springer, New York.

Heij, C., Groenen, P.J., van Dijk, D., 2007. Forecast comparison of principal component regression and principal covariate regression. Comput. Stat. Data Anal. 51, 3612–3625.

Hoerl, A.E., Kennard, R.W., 1970. Ridge regression: biased estimation for nonorthogonal problems. Technometrics 12, 55–67.

Kaiser, H.F., 1958. The varimax criterion for analytic rotation in factor analysis. Psychometrika 23, 187–200.

McNeish, D.M., 2015. Using lasso for predictor selection and to assuage overfitting: a method long overlooked in behavioral sciences. Multivar. Behav. Res. 50, 471–484.

Ooi, H., 2021. glmnetUtils: utilities for 'Glmnet'. R package version 1.1.8. URL. https://CRAN.R-project.org/package=glmnetUtils.

Putka, D.J., Beatty, A.S., Reeder, M.C., 2018. Modern prediction methods: new perspectives on a common problem. Organ. Res. Methods 21, 689–732.

R Core Team, 2020. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/.

Rosseel, Y., 2012. lavaan: an R package for structural equation modeling. J. Stat. Softw. 48, 1–36. https://doi.org/10.18637/jss.v048.i02.

Schmid, M.S., Dusseldorp, E., 2010. Quantitative analyses in a multivariate study of language attrition: the impact of extralinguistic factors. Second Lang. Res. 26, 125–160.

Seeboth, A., Mõttus, R., 2018. Successful explanations start with accurate descriptions: questionnaire items as personality markers for more accurate predictions. Eur. J. Pers. 32, 186–201.

Skrondal, A., Laake, P., 2001. Regression among factor scores. Psychometrika 66, 563–575.

Thomson, G.H., 1935. The definition and measurement of "g" (general intelligence). J. Educ. Psychol. 26, 241.

Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. J. R. Stat. Soc., Ser. B, Methodol., 267–288.

Varma, S., Simon, R., 2006. Bias in error estimation when using cross-validation for model selection. BMC Bioinform. 7, 91.

Vervloet, M., Kiers, H.A., Van den Noortgate, W., Ceulemans, E., 2015. PCovR: an R package for principal covariates regression. J. Stat. Softw. 65, 1–14. http://www.jstatsoft.org/v65/i08/.

Vervloet, M., Van Deun, K., Van den Noortgate, W., Ceulemans, E., 2016. Model selection in principal covariates regression. Chemom. Intell. Lab. Syst. 151, 26–33.

Zou, H., Hastie, T., 2005. Regularization and variable selection via the elastic net. J. R. Stat. Soc., Ser. B, Stat. Methodol. 67, 301–320.